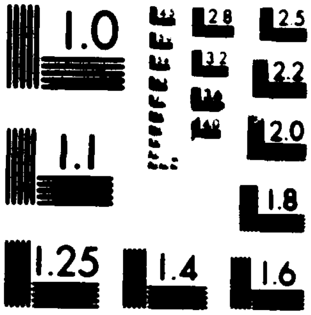
AD-A182 711 THE BOX METHOD FOR LINEAR PROGRAMMING PART I BASIC THEORY(U) STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB K ZIKAN ET AL. JUN 87 SOL-87-6 N00014-85-K-0343 1/1

UNCLASSIFIED F/G 12/4 NL

END
DATE
FILMED
8[illegible]


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

Systems
Optimization
Laboratory

# The Box Method for Linear Programming: Part I — Basic Theory

by
Karel Zikan
Richard W. Cottle

TECHNICAL REPORT SOL 87-6

June 1987

Department of Operations Research
Stanford University
Stanford, CA 94305

12

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

# The Box Method for Linear Programming: Part I — Basic Theory

by

Karel Zikan
Richard W. Cottle

TECHNICAL REPORT SOL 87-6

June 1987

Research and reproduction of this report were partially supported by the National Science Foundation Grant DMS-8420623, U.S. Department of Energy Grant DE-FG03-87-ER25028; Office of Naval Research Contract N00014-85-K-0343.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do NOT necessarily reflect the views of the above sponsors.

# THE BOX METHOD FOR LINEAR PROGRAMMING: PART 1 – BASIC THEORY

by Karel ZIKAN and Richard W. COTTLE

## 1. Introduction.

Interior-point linear programming algorithms have been described as methods in which "one plunges into the interior of the feasible region" [Kolata (1984)] instead of "crawling along its edges" [Brown and Koopmans (1951)]. One way or another, ellipsoids play an important role in both well known interior-point algorithms for linear programming. Their role is quite conspicuous in the so-called ellipsoid method of Hacijan (1979), whereas their role is more subtle in the projective scaling algorithm of Karmarkar (1984a), (1984b) and its affine scaling variants, e.g., Adler, Resende, and Veiga (1986), Barnes (1986), Vanderbei, Meketon, and Freedman (1986). Indeed, Adler, Resende, and Veiga (op.cit.) note solving the direction-finding subproblem is tantamount to optimizing the main problem's linear objective function over an ellipsoid. It is generally believed that the direction-finding subproblem can be a heavy computational burden in an otherwise very efficient method. The work reported here was inspired by efforts to reduce this computational burden.

Unlike the aforementioned interior-point algorithms, the one presented in this paper *does not* explicitly or implicitly use ellipsoids in its direction-finding scheme. Instead, this algorithm uses "boxes", or, more precisely, parallelepipeds.[1] Each such box is associated with a basic solution of the constraints of the problem. The computation of each search direction is done over the current box. These subproblems are linear programs having closed form solutions. Accordingly, this interior-point linear programming algorithm makes no use of nonlinear programming.

The algorithm presented here works with linear programs whose constraints are expressed as linear inequalities. It is initiated at an interior feasible point which is assumed to be available. A way to obtain such a point, if one exists, is discussed in Adler, Resende, and Veiga (1986) and Murty (1986). The problem formulation, relevant assumptions, and motivation for the algorithm are set forth in Section 2. The algorithm is stated in Section 3 and shown to converge in Section 4. As a by-product of the methodology introduced here, one obtains a finite subdivision of $R^n$; it is presented in Section 5. This much will complete the basic theory of the box method for general linear programs.

[1] In light of how difficult it is to pronounce this word, to say nothing of spelling it, we hesitate to propose the name "parallelepiped algorithm."

A specialization of the algorithm to minimal cost network flows and in particular to the dual of the standard transportation problem, as well as implementation issues and computational experience for the general linear programming case will be discussed in subsequent reports.

## 2. Formulation, Assumptions, and Motivation.

The algorithm described in this paper is designed for linear programs of the form

$$\begin{aligned} &\text{maximize} \quad c^{\mathrm{T}}x && (1.1)\\ &\text{subject to} \quad Ax \le b && (1.2)\end{aligned}$$

Such a problem is dual to one of the (so-called) standard form

$$\begin{aligned} &\text{minimize} \quad b^{\mathrm{T}}y && (2.1)\\ &\text{subject to} \quad A^{\mathrm{T}}y = c && (2.2)\\ &\qquad\qquad y \ge 0 && (2.3)\end{aligned}$$

Accordingly, the algorithm may (but need not) be thought of as a "dual method." Let

$$X = \{x : Ax \le b\}. \tag{3}$$

Regarding the data of (1), we make the following assumptions:

(A1) $A \in R^{m\times n}, b \in R^m, c \in R^n, m \ge n;$

(A2) $\text{rank}(A) = n;$

(A3) $\|A_{i\cdot}\|_2 = 1 \quad i = 1, \ldots, m;$

(A4) $\exists x^0 \in R^n$ such that $Ax^0 < b;$

(A5) $\{x \in X : c^{\mathrm{T}}x \ge c^{\mathrm{T}}x^0\}$ is bounded.

Assumption (A2) is not absolutely necessary, but it greatly simplifies the exposition of the method. Assumption (A3), which says that the rows of $A$ have unit length in the Euclidean norm, can be made to hold unless $A$ has a row of zeros, which would be quite abnormal. Assumption (A4), while not automatic, is not unusual, either. Initialization schemes have been proposed in the literature. This assumption implies that $x^0$ is an *interior point* of $X$. The inclusion of assumption (A5) is undesirable, but not unusual.

For any vector $\bar{x} \in X$, define the nonnegative (slack) vector

$$w(\bar{x}) = b - A\bar{x}. \tag{4}$$

In light of the normalization assumption (A3), the components of $w_i(\bar{x})$ can be interpreted as the Euclidean distances to the corresponding *hyperplanes*

$$H_i = \{x : \sum_{j=1}^{n} a_{ij}x_j = b_i\}. \tag{5}$$

When $\bar{x} \in \text{int}X$, these distances are all positive. For such a vector, one can define a diagonal matrix

$$D = \text{Diag}(1/\bar{w}_1, \ldots, 1/\bar{w}_m) \tag{6}$$

where $\bar{w} = w(\bar{x})$.

The key direction-finding subproblem in a standard variant of Karmarkar's algorithm entails solving the equation

$$A^{\mathrm{T}}D^2Ad_x = c, \tag{7}$$

and then taking

$$d_w = -Ad_x. \tag{8}$$

The *search direction* $d_x$ given by the solution to (7) can be interpreted as the solution to the problem of maximizing the linear form $c^{\mathrm{T}}z$ over the ellipsoid

$$Z = \{z : z^{\mathrm{T}}A^{\mathrm{T}}D^2Az \leq \kappa\} \tag{9}$$

(where $\kappa$ is a positive constant). One can think of this ellipsoid as being centered at a current iterate and contained within the polyhedron $X$. This leads to the notion that the direction given by a suitable *approximation* to the aforementioned ellipsoid may produce a direction vector that is cheaper to compute yet nearly as good as the one given by (7). This is where the interpretation of the slack variables as distances to the bounding hyperplanes becomes important. It is at least intuitively clear that the shape of an ellipsoid determined by an interior point close to a vertex will not be strongly influenced by the more remote bounding hyperplanes, i.e., those for which $w_i$ is large. This suggests trying to use a basis $B$ in $A$ that corresponds to a set of "nearby" hyperplanes.

To find such a matrix, consider the distances (slack variables) $\bar{w}_i = w_i(\bar{x})$ corresponding to the interior point $\bar{x}$. There will exist an index set $E$ of cardinality $n$ such that the vectors $\{A_{i\bullet} : i \in E\}$ are *linearly independent* and $\sum_{i \in E} \bar{w}_i$ is *minimum*. A matrix of the type $A_{E\bullet}$ (the matrix consisting of the rows $A_{i\bullet}$ for all $i \in E$) will be called a *minimum-weight basis* corresponding to $\bar{x}$. For a given point $\bar{x}$ there may be several such sets $E$. Each of these sets determines a unique point $x(E)$ which may or may not belong to $X$. If $x(E)$ belongs to $X$, then it must be an extreme point.

Instead of using an ellipsoid $Z$ as in (9) or even an ellipsoidal approximation to $Z$ for calculating a search direction at $\bar{x}$, consider the parallelepiped

$$P(E) = \{z : -\bar{w}_E \leq A_{E\cdot}z \leq \bar{w}_E\}. \tag{10}$$

Likewise, $\bar{w}_E$ is the subvector of $\bar{w}$ corresponding to $E$. For brevity, denote the $n \times n$ nonsingular matrix $A_{E\cdot}$ by $B$.

The "box problem" corresponding to $\bar{x}$ is

$$\begin{aligned} &\text{maximize} && c^{\mathrm{T}}z && (11.1)\\ &\text{subject to} && Bz \leq \bar{w}_E && (11.2)\\ & && -Bz \leq \bar{w}_E && (11.3) \end{aligned}$$

Its feasible region is just the set $P(E)$ defined in (10).

To see that the box pro'blem (11) can be solved in closed form, let

$$c^{\mathrm{T}} = \tilde{c}^{\mathrm{T}}B, \tag{12}$$

and

$$\tilde{z} = Bz. \tag{13}$$

Then (11) can be written as

$$\begin{aligned} &\text{maximize} && \tilde{c}^{\mathrm{T}}\tilde{z} && (14.1)\\ &\text{subject to} && \tilde{z} \leq \bar{w}_E && (14.2)\\ & && -\tilde{z} \leq \bar{w}_E && (14.3\text{\textbackslash} \end{aligned}$$

To form (14.1) one needs $\tilde{c}^{\mathrm{T}}$ which can be gotten from (12) through an L-U factorization of $B$. Suppose

$$E = \{i_1, \ldots, i_n\}. \tag{15}$$

For $j = 1, \ldots, n$, let

$$\begin{aligned} \tilde{z}_j &= \bar{w}_{i_j} \quad \text{if } \tilde{c}_j \geq 0, && (16.1)\\ \tilde{z}_j &= -\bar{w}_{i_j} \quad \text{if } \tilde{c}_j < 0. && (16.2) \end{aligned}$$

It is clear that the vector $\tilde{z}$ defined by (16) is optimal for (14). The same factorization of the matrix $B$ can be used in (13) to construct an optimal solution of (11) from that of (14). Thus, the box problem has a closed form solution.

For each $i \in E$, the set $P(E)$ has two sides (facets) parallel to the hyperplane $H_i$ given by (5). The vertex of $P(E)$ determined by $z = B^{-1}\bar{w}_E$ is analogous to the point $x(E)$ mentioned above. In fact, $x(E) = \bar{x} + z$. The hyperplanes that determine $z$ are parallel to (a subset of) those incident to $x(E)$. Thus, the set $\bar{x} + P(E)$ fits "neatly" into the corner at $x(E)$. If $x(E)$ is nondegenerate, the fit is exact. If $x(E) \in X$, and $z = B^{-1}\bar{w}_E$ solves the box problem at $\bar{x}$, then moving from $\bar{x}$ in the direction $z$ with a step length of 1 yields an *optimal solution* of the original problem (1).

## 3. Statement of the Algorithm.

This section presents the algorithm in its general form for linear programming purposes. Specializations will be treated in later reports.

The input to the algorithm consists of a linear program in the form (1) with data satisfying (A1)–(A5), and a real number $\theta$ such that $0 < \theta < 1$.

### Algorithm I: The Box Method

**Step 0.** Determine $x^0 \in \operatorname{int} X$. Set $k = 0$.

**Step 1.** Given $x^k \in \operatorname{int} X$ compute $w^k := b - Ax^k$.

**Step 2.** Determine a minimum-weight basis $A_{E\cdot}$ corresponding to $x^k$.

**Step 3.** Compute $z^k$, the solution of (11), via (12), (13), (15), and (16).

**Step 4.** Define:

$$q^k := Az^k; \tag{17}$$

$$\rho_k := \min\{w_i^k/q_i^k : q_i^k > 0,\ i = 1, \ldots, m\}; \tag{18}$$

$$\sigma_k := \theta\rho_k. \tag{19}$$

**Step 5.** If $\rho_k = 1$ and $\bar{c}^{\mathrm{T}} = c^{\mathrm{T}}A_{E\cdot}^{-1} \geq 0$, stop. The vector $x^{k+1} = x^k + z^k$ is optimal. Otherwise, define

$$x^{k+1} = x^k + \sigma_k z^k. \tag{20}$$

If the termination criterion is satisfied, stop; construct an optimal solution. Otherwise, replace $k$ by $k + 1$ and return to Step 1.

### Comments on Algorithm I.

1. *Transversal step.* The initial interior point $x^0$ may be deemed "too far" from the boundary of $X$, and it may be advantageous to "reinitialize" the algorithm by finding

another interior point such as one of the form

$$x^0 + \theta\rho c \tag{21}$$

where

$$q = Ac \tag{22.1}$$
$$\rho = \min\{w_i^0/q_i : q_i > 0,\ i = 1,\dots,m\}. \tag{22.2}$$

The point given by (21) yields a larger objective function value than $x^0$ does. Indeed,

$$c^{\mathrm{T}}(x^0 + \theta\rho c) - c^{\mathrm{T}}x^0 = \theta\rho\|c\|^2 \tag{23}$$

could be large. The transversal step (21) is analogous to the opening move in Murty's gravitational method for linear programming. See Murty (1986). The idea of moving all the way to the boundary in the gradient direction dates back at least as far as the 1951 paper of Brown and Koopmans.

2. *Finding a minimum weight basis.* The work to be done in Step 2 entails a partial sorting of the slack variables $w_i^k$. In the most favorable case, the $n$ smallest components of $w^k$ correspond to linearly independent rows of $A$ and thereby furnish the minimum-weight basis $A_{E.}$. When these $n$ rows are linearly dependent, something special must be done. A practical algorithm for executing this step will be given in a forthcoming report on implementation details and computational experience. At present, it suffices to note that finding a minimum weight basis is a matroidal problem solvable by a greedy algorithm. [See Papadimitriou and Steiglitz (1982), pp. 280-289.]

3. *Use of asymmetric boxes.* The parallelepiped $P(E)$ defined in (10) and used in Step 3 is symmetric about the origin. It is not really necessary to use a parallelepiped having this symmetry property. It seems plausible that allowing "more room" toward the interior of $X$ could be advantageous from the computational standpoint. Thus, for example, one could define

$$P_\mu = \{z : -\mu\bar{w}_E \le A_{E.}z \le \bar{w}_E\},\ \mu \ge 1 \tag{24}$$

thereby making $P(E) = P_1(E)$. The important point is to capture some of the "shape" of the set formed by the intersection of the halfspaces whose bounding hyperplanes are the $H_i$, $i \in E$. The technique for solving the box problem over $P_\mu$ is much the same as given in Section 2 and will not be repeated here. See Figure 1.

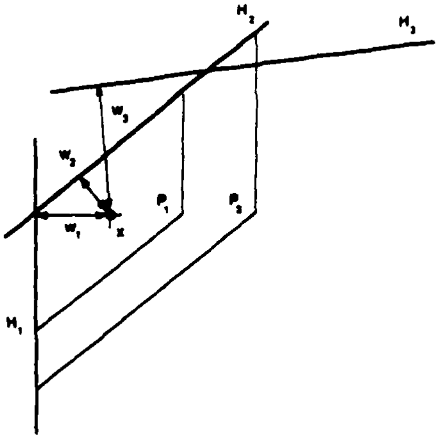


Figure 1: Asymmetric boxes.

4. *Step size.* The calculation of $\rho_k$ in Step 4 determines how far it is possible to move from $x^k$ in the direction $z^k$ before meeting the boundary. Since (as will be shown in Section 4) the objective function $c^{\mathrm{T}}x$ increases in the $z^k$ direction, the vector $q^k$ must have at least one positive component, for otherwise (A5) would be violated. The parameter $\theta$ keeps the new iterate in the interior of the feasible region $X$. Like $\mu$, value of this parameter is to be chosen by the user. It is common (in similar algorithms) to use a value close to 0.9. The computational implications of different choices of $\mu$ and $\theta$ will also be discussed in future reports.

5. *Optimality criteria.* It should be noted that in Step 5, the shrinking factor $\theta$ is not used when the two conditions $\rho_k = 1$ and $\tilde{c}^{\mathrm{T}}A_{E\cdot}^{-1} \geq 0$ obtain. In this instance, it is an easy exercise to show that $x^{k+1}$ is an optimal solution for (1). Moreover, $\tilde{c}$ is the vector of basic variables in an optimal basic feasible solution (corresponding to the basis $B^{\mathrm{T}}$) of the linear program (2). Thus, when thinking of (1) as the dual problem, it is a simple matter to reconstruct a solution of the primal problem from information at hand when a solution of the dual has been found.

To handle the situation where termination must be induced, Step 5 requires a user-supplied convergence criterion. (See Gill, Murray, and Wright (1981) for some possibilities.) Being an interior-point, the final iterate so determined can be only approximately optimal. Passage to a "nearby" optimal vertex of $X$ can be facilitated by Gram-Schmidt orthogonalization[2] followed (if necessary) by application of the (dual) simplex algorithm to the standard form problem, (2). This procedure terminates with the usual pair of primal and dual optimal solutions.

[2] The vertex gotten this way is necessarily an improvement over the interior point from which it was derived.

## 4. Convergence.

This section is devoted to proving the convergence of Algorithm I. It will be shown that starting from any interior point, the algorithm generates a convergent sequence of iterates whose limit must be a vertex of the feasible region, $X$. It will also be shown that, when this vertex is *nondegenerate*, it must be an optimal solution of the linear program (1) and moreover must be reached after a finite number iterations. Since each iteration involves only finitely many operations, the algorithm is finite in this case. It is well known in the folklore that the feasible regions of most *real* linear programs have degenerate vertices, hence recognition and handling of this fact is an essential part of a theoretically satisfactory linear programming algorithm. These aspects of the algorithm will be addressed later. As a practical matter, the solution recovery procedure pertaining to Step 5 that was sketched in the fifth comment in Section 3 could be employed to deal with the case of termination at a suboptimal solution.

The convergence proof will be developed in a series of lemmas. Most of them will use the key idea of the *constancy* of a certain "significant basis." The purpose of the next two paragraphs is to define this notion and the related concept of "significant direction."

The $m \times n$ matrix $A$ has rank $n$, hence it has at least one $n \times n$ nonsingular submatrix. Any such matrix is called a *basis* in $A$; for a given matrix, the set of all bases in $A$ will be denoted $\mathcal{B}$. For each member $B$ of $\mathcal{B}$, there is a corresponding set $E$ of row indices such that $B = A_{E\cdot}$. This correspondence makes it possible to identify basis matrices with basis index sets, and to speak of the latter as bases—indeed, to speak of these index sets as the elements of $\mathcal{B}$.

Recall that for each $x \in X$, there is a slack vector $w(x) = b - Ax$. Given such an $x$, let $\mathcal{B}(x)$ denote the set of all $E \in \mathcal{B}$ such that

$$\sum_{i \in E} w_i(x) \leq \sum_{i \in E'} w_i \qquad \text{for all } E' \in \mathcal{B}$$

The set $\mathcal{B}(x)$ is clearly nonempty. Its elements are called *significant bases* at $x$. This set is well defined for all points in $R^n$ and, in particular, all points of $X$.

Given $x \in X$, let $\mathcal{Z}(x)$ be the set of all directions that could be chosen as solutions to the box problem in accordance with Algorithm I. That is, .

$$\mathcal{Z}(x) = \bigcup_{E \in \mathcal{B}(x)} \{z : z \text{ solves (11) via (12), (13), (15), and (16)}\}$$

By making suitable and obvious modifications in the definition of the box problem, one can extend the definition of $\mathcal{Z}(x)$ to all points of $X$. The set $\mathcal{Z}(x)$ contains the zero vector if and only if $x$ is a vertex of $X$, in which case 0 is the only member of $\mathcal{Z}(x)$. Thus, when $x$ is not a vertex of $X$, the members of $\mathcal{Z}(x)$ are called *significant directions* at $x$.

The following lemma deals with significant bases of points in the feasible region $X$, but not necessarily with Algorithm I.

**Lemma 1.** *For every $x \in X$ there exists an epsilon neighborhood $N(x)$ such that*

$$B(x) \supseteq \cup\{B(z) : z \in N(x) \cap X\}. \tag{25}$$

**Proof.** Suppose there is a point $\bar{x} \in X$ for which no such neighborhood exists. Then there is a sequence $\{\bar{x}^k\}$ converging to $\bar{x}$ such that for each $\bar{x}^k$ there is a corresponding significant basis not belonging to $B(\bar{x})$. (These bases could then be called "insignificant" at $\bar{x}$.) Since there are only finitely many bases in $A$ (i.e., $B$ is finite), there is a subsequence of $\{\bar{x}^k\}$ for which the aforementioned insignificant basis at $\bar{x}$ is the same for every member of the subsequence. Let this basis be $E'$. Thus, $E' \notin B(\bar{x})$. Let $E^*$ denote any element of $B(\bar{x})$. Now

$$\sum_{i \in E'} w_i(\bar{x}^k) \leq \sum_{i \in E^*} w_i(\bar{x}^k) \qquad \text{for all } k \text{ (in the sequence)}, \tag{26}$$

and, because $E'$ is not a significant basis at $\bar{x}$,

$$\lim_k \sum_{i \in E'} w_i(\bar{x}^k) = \sum_{i \in E'} w_i(\bar{x}) > \sum_{i \in E^*} w_i(\bar{x}). \tag{27}$$

A contradiction can now be obtained by taking limits in (26) and combining the resulting inequality with the one in (27). ■

**Remark.** At any iteration $k$ of Algorithm I, the current point $x^k$ belongs to the interior of $X$. Likewise, the origin is interior to the box $P(E)$. From this (and the natural assumption that $c$ is nonzero) it follows that $c^T z^k > 0$ for any optimal solution of the box problem. Thus, with $x^{k+1} = x^k + \sigma_k z^k$, it is clear that Algorithm I has the *strict ascent property*

$$c^T x^{k+1} = c^T x^k + \sigma_k c^T z^k > c^T x^k. \tag{28}$$

By virtue of (28) and (A5), the iterates $x^k$ lie in a compact set. Accordingly, the sequence $\{x^k\}$ must have at least one cluster point. Furthermore, since $z^k \neq 0$ for all $k$, it follows that the sequence of normalized direction vectors $\hat{z}^k = (1/\|z^k\|)z^k$ must have at least one cluster point.

**Lemma 2.** *If $\{x^k\} \to x$ and $\{z^k\} \to z$ are convergent (sub)sequences of iterates and corresponding significant directions generated by Algorithm I, then $z \in Z(x)$.*

**Proof.** By taking subsequences if necessary, one can assume that the $z^k$ are all solutions to box problems formed by the *same* basis, say $E$. This means that the matrix $B = A_{E\cdot}$ does not change. By Lemma 1, $E$ must belong to $\mathcal{B}(x)$. Thus, the vector $\tilde{c}$ in (12) does not change either. Since $w(x^k) \to w(x)$, it now follows from (16) that $\tilde{z}^k \to \tilde{z}$. Thus,

$$\tilde{z} = \lim_k \tilde{z}^k = \lim_k A_{E\cdot} z^k = A_{E\cdot}(\lim_k z^k) = A_{E\cdot} z,$$

and in particular $z$ satisfies (13). It now follows that $z$ solves the box problem associated with $x$ in accordance with Step 3 of Algorithm I. This means $z \in Z(x)$. ■

**Lemma 3.** *If $x^k$ is a convergent (sub)sequence of vectors generated by Algorithm I and $\{\rho_k\}$ is the corresponding sequence of step-sizes, then*

$$\rho^* = \liminf \rho_k > 0. \tag{29}$$

**Proof.** The sequences can be refined to subsequences in such a way that the step-size sequence converges to $\rho^*$ and the significant bases for the iterates $x^k$ are all the same, say $E$. Let $z^k$ be the corresponding subsequence of significant directions and let $q^k$ be defined as in (17), while $\rho_k$ is defined as in (18). The sequences can be refined still further so that the same subscript, say $h$, occurs as the minimizer in (18). That is, $\rho_k = w_h^k / q_h^k$ for all $k$. Clearly

$$\lim_k w_h^k > 0 \implies \lim_k w_h^k / q_h^k = w_h / q_h > 0.$$

Also, if $h \in E$, then $q_h^k = w_h^k$ for all $k$ and $\lim_k w_h^k / q_h^k = 1 > 0$. Therefore $w_h^k \to 0$ and $h \notin E$ are two *necessary conditions* for the subsequence of $\rho_k$ to converge to 0. The remainder of the proof amounts to showing that these two conditions lead to a contradiction.

For any vector $\hat{x} \in R^n$, let $I_0(\hat{x}) = \{i : w_i(\hat{x}) = 0\}$. Now suppose $w_h^k \to 0$ and $h \notin E$. Since $E$ is a basis, the row vector $A_{h\cdot}$ is linearly dependent on the rows of $A_{E\cdot}$. Recall that $\lim_k x^k = x$. It follows that $h \in I_0(x)$. Moreover, $A_{h\cdot}$ is linearly dependent on the rows of $A$ indexed by $E \cap I_0(x)$, for otherwise $h$ would have been chosen as an element of $E$. It can be shown that the index set $E \cup \{h\}$ contains a unique minimum cardinality subset $C$ for which the corresponding rows of $A$ are linearly dependent. One of these indices must be $h$. (In the language of matroids, such a set would be called a *circuit*. See Papadimitriou and Steiglitz (1982).)

Each significant direction vector $z^k$ can be decomposed into the direct sum of orthogonal components $\bar{z}^k$ and $\bar{\bar{z}}^k$ where $\bar{z}^k$ lies in the subspace spanned by the normals of the hyperplanes whose indices lie in $C$. These $\bar{z}^k$ are the "significant components" of the direction vectors in the sense that

$$(Az^k)_i = (A\bar{z}^k)_i \text{ for all } i \in C.$$

For each $k$ in the sequence, define

$$\alpha^k = \max_{i \in C} w_i^k.$$

This maximum must be achieved when $i = h$, for otherwise $w_h^k < w_j^k$ for some $j \in C$. Then after a replacement in $E$ of $j$ by $h$, a new and "lower-weight" basis $E'$ would be obtained, in contradiction to the defining property of $E$. Thus, $\alpha^k = w_h^k$.

To obtain a bound on $\|\bar{z}^k\|$, recall that $\bar{z}^k$ lies in the space spanned by $A_{C\cdot}$ and

$$(Az^k)_i = (A\bar{z}^k)_i = \tilde{z}_i^k \quad \text{for all } i \in C \setminus \{h\} \tag{30}$$

where $|\tilde{z}_i^k| = w_i^k$ (as in (16)). Assume the cardinality of $C$ is $L+1$. All the relevant vectors lie in an $L$-dimensional subspace. After an isometric transformation of the subspace to $R^L$, the system (30) becomes one of the form

$$Mu^k = y^k,$$

where $\|u^k\| = \|\bar{z}\|$ and $\|y^k\| = \|\tilde{z}^k\|$. From (17) and (A3) one gets

$$q_h^k = A_{h\cdot}\bar{z}^k \le \|A_{h\cdot}\|\,\|\bar{z}^k\| = \|\bar{z}^k\|\,;$$

it then follows that

$$\rho_k = \frac{w_h^k}{q_h^k} \ge \frac{1}{\sqrt{L}\|M^{-1}\|} > 0.$$

This inequality holds for all $k$. Notice that the quantity $1/\sqrt{L}\|M^{-1}\|$ is independent of $k$ and hence provides a positive uniform lower bound on $\rho_k$. ■

**Lemma 4.** *If $\{x^k\}$ is a sequence of iterates generated by Algorithm I and $x$ is a cluster point of $\{x^k\}$, then $x$ is an extreme point of $X$.*

**Proof.** Suppose $x$ is not an extreme point of $X$. Then each element $z$ of $Z(x)$ is a nonzero ascent direction:

$$c^{\mathrm{T}}x < c^{\mathrm{T}}(x + \lambda z) \text{ for all } \lambda > 0.$$

Let $\{z^k\}$ be the corresponding sequence of significant directions. By extracting a suitable subsequence (if necessary) one can assume $x^k \to x$ and $z^k \to z$. This is possible since

Lemma 3 implies that $\|z^k\|$ cannot get arbitrarily large for $x^k$ in some epsilon neighborhood $N(x)$ of the point $x$. Therefore $\{z^k\}$ must have a cluster point.

Now, for the subsequence $\{x^k\}$ as just constructed, consider the sequence $\{y^k\}$ of their *algorithmic successors*, that is, the points of the form

$$y^k = x^k + \sigma_k z^k.$$

Also,

$$\liminf c^{\mathrm{T}} y^k \geq c^{\mathrm{T}}[x + \theta(\liminf \rho_k) z] > c^{\mathrm{T}} x.$$

Thus, $c^{\mathrm{T}} y^k > c^{\mathrm{T}} x$ for almost all $k$. This implies $c^{\mathrm{T}} y^{k-1} > c^{\mathrm{T}} x$ for almost all $k$. It now follows that

$$c^{\mathrm{T}} x^k < c^{\mathrm{T}} x < c^{\mathrm{T}} y^{k-1} \leq c^{\mathrm{T}} x^k \quad \text{for almost all } k,$$

and this is a contradiction. ■

**Lemma 5.** *The sequence of iterates $x^k$ generated by Algorithm I converges to a vertex of $X$.*

**Proof.** Since the entire sequence lies in a compact set, it must have at least one cluster point. In Lemma 4, it was shown that any such cluster point is a vertex of $X$. Thus, it remains to show that $\{x^k\}$ has only one cluster point.

Suppose $\{x^k\}$ has two (or more) cluster points, say $v^1, \ldots, v^p$. Let $N(v^1), \ldots, N(v^p)$ be disjoint epsilon neighborhoods of the sort described in Lemma 1. The union of these neighborhoods contains all but a finite number of the $x^k$. Since there are finitely many vertices of $X$, it follows that (after relabeling if necessary) for infinitely many values of $k$, $x^k \in N(v^1)$ and $x^{k+1} \in N(v^2)$. Let $\{x^\ell\}$ denote the first of these subsequences and let $\{y^\ell\}$ denote the other. Thus, $x^\ell \to v^1$ and $y^\ell \to v^2$. Note that $y^\ell = x^\ell + \sigma_\ell z^\ell$. Define

$$u^\ell = x^\ell + \rho_\ell z^\ell.$$

Since $\{u^\ell\}$ is a sequence of feasible points, all its cluster points belong to $X$. (There must be at least one such point.) Without loss or generality, one can assume that $\{u^\ell\} \to u$. It follows from these definitions that

$$y^\ell = (1 - \theta) x^\ell + \theta u^\ell.$$

After taking limits, one has

$$v^2 = (1-\theta)v^1 + \theta u. \tag{31}$$

Clearly $u \neq v^1$, for $v^2 \neq v^1$. Since $0 < \theta < 1$, equation (31) states that the extreme point $v^2$ is a proper convex combination of distinct points of $X$. This is a contradiction. ■

**Theorem.** *Given $x^0 \in \text{int}X$, let $\{x^k\}$ denote the sequence of points generated by Algorithm I, and let $x$ denote the extreme point of $X$ to which the sequence converges. If $x$ is nondegenerate, then:*

(*i*) *$x$ is an optimal solution of* (1);

(*ii*) *the algorithm terminates after finitely many iterations.*

**Proof.** Since $x$ is a nondegenerate vertex, it corresponds to exactly one basis, $E$. As the iterates approach $x$, they enter an epsilon neighborhood $N(x)$ of the sort for which (according to Lemma 1) the inclusion (25) holds. In other words, the significant bases of the iterates become (and remain) the same as the one for $x$, namely $E$. Thus, if the vertex $X$ is not optimal, the solution to (12) must have at least one negative component, say $\tilde{c}_i$. This implies $(Az^k)_i = -w_i^k$, indicating that $w_i$ should continually increase (instead of converging to 0). This contradiction comple'es the proof of assertion ($i$).

Assertion ($ii$) follows from the fact that after finitely many steps, the iterates reach the aforementioned neighborhood $N(x)$ of the extreme point $x$. From any feasible point of this neighborhood, the algorithm will detect the optimality of $x$. In particular, if $x^k \in N(x)$, then $x^{k+1} = x$ because $\rho_k = 1$ and $\tilde{c} \geq 0$. ■

## 5. Minimum-weight subdivisions.

It is the purpose of this section to show that for any polyhedron $X$ satisfying assumptions (A1)-(A4), the concept of a minimum-weight basis (as defined in Section 2) leads to a finite subdivision of $X$, and indeed to the entire space $R^n$. Concerning subdivisions, see Eaves (1976).

Figure 2 below illustrates this assertion. In this example, there are five lines, no two of which are parallel. These lines have pairwise intersections in ten points, six of which are labeled, $v_1, \ldots, v_6$. The polyhedron $X$ in question is a quadrilateral, the convex hull of $\{v_1, v_2, v_3, v_4\}$. Each vertex of $X$ is nondegenerate, being the intersection of exactly two lines. The other two labeled points, $v_5$ and $v_6$ are nondegenerate, but are not members of $X$. The four remaining points are also nondegenerate. To each labeled point $v_i$ there corresponds a polyhedral subset of $X$ labeled $i$ throughout (the interior of) which the points have the same significant basis and hence correspond to the intersection of the same lines, namely $v_i$. These polyhedral subsets (called *cells*) form a subdivision of $X$. The cell corresponding to each vertex of $X$ contains that vertex as well as points in its neighborhood. As Figure 2 shows, these four cells do not fill up $X$ as cells 5 and 6 do not correspond to vertices of $X$.

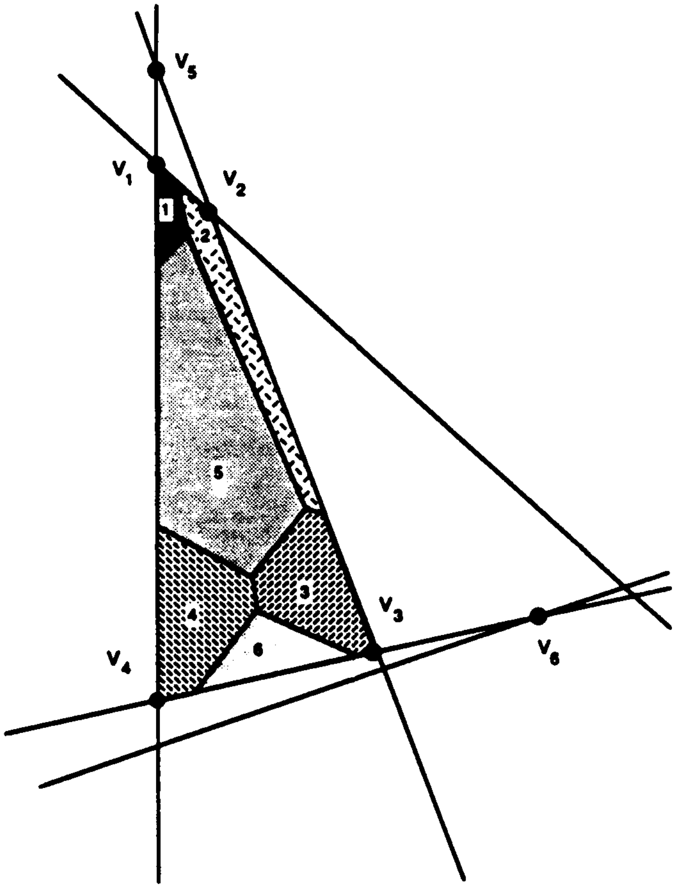


Figure 2: Minimum-weight subdivision of a polyhedron.

The other polyhedral regions of the plane are the solution sets of linear inequality systems based on the same data with the sense of one or more inequalities reversed, i.e., $\leq$ replaced by $\geq$.

Suppose the polyhedron $X$ in Figure 2 is the feasible region of a linear program (1). At each interior point of one of the six cells of $X$ the box problem formed in Algorithm I is the same. A point of $X$ belonging to the intersection of two or more cells has two or more significant bases.

In Figure 2, the unique point (call it $x$) belonging to cells 4, 5, and 6 illustrates Lemma 1. Within a sufficiently small epsilon neighborhood of this point, the set of significant bases of every point $z$ is a significant basis of $x$. If the epsilon neighborhood were too large, it would contain points for which the basis that gives rise to $v_6$ is significant but that basis is not significant for this $x$.

The following discussion is aimed at generalizing these observations. The goal is to show that the notion of minimum-weight basis induces a subdivision of $R^n$. Many of the definitions and notations are the same as those in Section 4, but the objective function of the original linear programming is of no immediate consequence for this purpose.

Let $X$ be the set defined in (3) where $A$ and $b$ satisfy the assumptions stated in (A1)-(A4). In principle, the $w$ vector in a solution $(x, w)$ of the system equations

$$Ax + w = b \tag{32}$$

could belong to any of $2^m$ orthants. Requiring that $w$ belong to one of these orthants is equivalent to specifying a linear inequality system

$$A_{i.}x \diamond b_i \quad i = 1, \dots m$$

where $\diamond$ stands for $\leq$ or $\geq$. Each such system corresponds to a polyhedral set in $R^n$. Some of these sets may be empty, but all the nonempty ones must have interior points, as $X$ does.

For each point $x \in R^n$, it is still meaningful to define the *combinatorial optimization problem*

$$\text{COP}(x) \qquad \text{minimize}\,\{\sum_{i \in E} w_i(x) : E \in \mathcal{B}\}. \tag{33}$$

Since $\mathcal{B}$ is finite, COP($x$) has at least one optimal solution. As before, denote the set of these optimal solutions by $\mathcal{B}(x)$. An element of $\mathcal{B}(x)$ is called a *minimum-weight basis.*

**Definition.** If $x$ and $y$ belong to $R^n$, then

$$x \sim y \iff \mathcal{B}(x) = \mathcal{B}(y). \tag{34}$$

**Lemma 6.** *The relation* $\sim$ *defined in* (34) *is an equivalence relation.*

**Proof.** The assertion is obvious because equality is itself an equivalence. ■

**Remark.** The lemma would still be valid if equality in (34) were replaced by any equivalence relation defined on the set of all subsets of $\mathcal{B}$.

**Definition.** If $x$ and $y$ belong to $R^n$, then

$$y \preceq x \iff \mathcal{B}(x) \subseteq \mathcal{B}(y). \tag{35}$$

**Lemma 7.** *The relation* $\preceq$ *defined in* (35) *is a partial ordering.*

**Proof.** Obvious. ■

**Lemma 8.** *For all* $x \in R^n$, *the set*

$$T(x) = \{y : y \preceq x\} \tag{36}$$

*is a polyhedron.*

**Proof.** It must be shown that $T(x)$ is the solution set of a finite system of weak linear inequalities. To see this, note that a vector $y$ belongs to $T(x)$ if and only if

$$\sum_{i \in E'} w_i(y) \leq \sum_{i \in E} w_i(y) \text{ for all } E' \in \mathcal{B}(x) \text{ and all } E \in \mathcal{B}$$

The lemma now readily follows from the definition $w_i(y) = b_i - A_{i.}y$. ■

**Remark.** The equivalence class

$$Y(x) = \{y : y \sim x\} \tag{37}$$

is the relative interior of $T(x)$.

## References.

I. Adler, M.G.C. Resende, and G. Veiga, An implementation of Karmarkar's algorithm for linear programming, Technical Report ORC 86-8, Operations Research Center, Department of Industrial Engineering and Operations Research, University of California, Berkeley, May 1986.

E.R. Barnes, A variation on Karmarkar's algorithm for solving linear programming problems, *Mathematical Programming* 36 (1986) 174-182.

G.W. Brown and T.C. Koopmans, Computational suggestions for maximizing a linear function subject to linear inequalities, in (T.C. Koopmans, ed.) *Activity analysis of production and allocation* (John Wiley & Sons, New York, 1951), 374-376.

B.C. Eaves, A short course in solving equations with *PL* homotopies, in (R.W. Cottle and C.E. Lemke, eds.) *Nonlinear Programming*, SIAM-AMS Proceedings, v. 9, (American Mathematical Society, Providence, RI, 1976) 73-143.

P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, (Academic Press, London, 1981.)

L.G. Hacijan, A polynomial algorithm in linear programming, *Doklady Akademii Nauk SSSR* 244 (1979) 191-194.

N. Karmarkar, A new polynomial-time algorithm for linear programming, *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing*, (1984a) pp. 302-311.

N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* 4 (1984b) 373-395.

G. Kolata, A fast way to solve hard problems, *Science* 225:4668 (21 September 1984) 1379-1380.

K.G. Murty, The gravitational method for linear programming, *Opsearch* 23 (1986) 206-214.

C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, (Prentice-Hall, Englewood Cliffs, NJ, 1982.)

R.J. Vanderbei, M.S. Meketon, B.A. Freedman, A modification of Karmarkar's linear programming algorithm, manuscript, *Algorithmica* 1 (1986) 395-408.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM |
| 1. REPORT NUMBER<br>SOL-87-6   2. GOVT ACCESSION NO.<br>A182711 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>The Box Method for Linear Programming:<br>Part I: Basic Theory | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical Report<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Karel Zikan and Richard W. Cottle | 8. CONTRACT OR GRANT NUMBER(s)<br>N00014-85-K-0343 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Department of Operations Research - SOL<br>Stanford University<br>Stanford, CA 94305 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>NR-047-064 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Office of Naval Research - Dept. of the Navy<br>800 N. Quincy Street<br>Arlington, VA 22217 | 12. REPORT DATE<br>June 1987<br>13. NUMBER OF PAGES<br>17 pp |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

| | |
|---|---|
| linear programming | interior-point algorithm |
| parallelepipeds | box constraints |
| minimum-weight basis | polyhedral subdivision |

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Please see attached.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOL 87-6: The Box Method for Linear Programming: Part I: Basic Theory
Karel Zikan and Richard W. Cottle

This paper presents a new interior-point algorithm for linear programming where the constraints are all expressed as inequalities. Along with the concept of "minimum-weight basis", the algorithm features a novel mechanism for finding search directions. Unlike other interior-point methods which implicitly or explicitly involve optimization over ellipsoids for their direction-finding schemes, the one reported here uses "boxes". The corresponding subproblems are simple linear programs having closed form solutions. It is shown that the iterates generated by the algorithm converge to an extreme point of the feasible region. When this point is nondegeneratə, it is optimal and reached within finitely any steps. The methodology introduced here also gives rise to a polyhedral subdivision of the problem's feasible region and in fact to the entire space of decision variables.